



```
PPPPPPPP      AAAAAA      SSSSSSSS      CCCCCCCC      LL      000000      SSSSSSSS      EEEEEEEEEE      222222
PPPPPPPP      AAAAAA      SSSSSSSS      CCCCCCCC      LL      000000      SSSSSSSS      EEEEEEEEEE      222222
PP      PP      AA      AA      SS      CC      LL      00      00      SS      EE      22      22
PP      PP      AA      AA      SS      CC      LL      00      00      SS      EE      22      22
PP      PP      AA      AA      SS      CC      LL      00      00      SS      EE      22      22
PP      PP      AA      AA      SS      CC      LL      00      00      SS      EE      22      22
PPPPPPPP      AA      AA      SSSSSS      CC      LL      00      00      SSSSSS      EEEEEEEE      22
PPPPPPPP      AA      AA      SSSSSS      CC      LL      00      00      SSSSSS      EEEEEEEE      22
PP      AAAAAAAAAA      SS      CC      LL      00      00      SS      EE      22
PP      AAAAAAAAAA      SS      CC      LL      00      00      SS      EE      22
PP      AA      AA      SS      CC      LL      00      00      SS      EE      22
PP      AA      AA      SS      CC      LL      00      00      SS      EE      22
PP      AA      AA      SSSSSSSS      CC      LL      00      00      SSSSSSSS      EEEEEEEEEE      2222222222
PP      AA      AA      SSSSSSSS      CCCCCCCC      LLLLLLLLLL      000000      SSSSSSSS      EEEEEEEEEE      2222222222
                                     ....
                                     ....
                                     ....
                                     ....

LL      I I I I I      SSSSSSSS
LL      I I I I I      SSSSSSSS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SSSSSS
LL      I I      SSSSSS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LLLLLLLLLLLL      I I I I I      SSSSSSSS
LLLLLLLLLLLL      I I I I I      SSSSSSSS
```

```
1 0001 0 MODULE PAS$CLOSE2 ( %TITLE 'close a file'
2 0002 0 IDENT = '1-006' ! File: PASCLOSE2.B32 Edit: SBL1006
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1 FACILITY: Pascal Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains PAS$CLOSE2 and PAS$$CLOSE, which close a file.
36 0036 1
37 0037 1 ENVIRONMENT: User mode - AST reentrant
38 0038 1
39 0039 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
40 0040 1
41 0041 1 MODIFIED BY:
42 0042 1
43 0043 1 1-001 - Original. SBL 1-April-1981
44 0044 1 1-002 - Give error in PAS$CLOSE2 if FCB is not valid. SBL 21-June-1982
45 0045 1 1-003 - Correct erroneous error check in PAS$$CLOSE. SBL 9-July-1982
46 0046 1 1-004 - Make sure XAB chain doesn't point to garbage. SBL 29-July-1982
47 0047 1 1-005 - Zero FCB$ STATUS before closing file. QAR FT3-2. SBL 30-Aug-1982
48 0048 1 1-006 - Use PAS$$AB KEYWD_NAME_TABLE to detect duplicate keywords.
49 0049 1 SBL 19-Aug-1983
50 0050 1 --
51 0051 1
```



PASSCLOSE2  
1-006

Close a file  
Declarations

I 14  
16-Sep-1984 01:29:07  
14-Sep-1984 12:51:21

VAX-11 Bliss-32 V4.0-742  
[PASRTL.SRC]PASCLOSE2.B32;1

Page 2  
(2)

```

53 0052 1 %SBTTL 'Declarations'
54 0053 1
55 0054 1 PROLOGUE DEFINITIONS:
56 0055 1
57 0056 1
58 0057 1 REQUIRE 'RTLIN:PASPROLOG';          ! Linkages, externals, PSECTs, structures
59 0121 1
60 0122 1
61 0123 1 TABLE OF CONTENTS:
62 0124 1
63 0125 1
64 0126 1 FORWARD ROUTINE
65 0127 1 PASSCLOSE2: NOVALUE,                ! Called by compiled code
66 0128 1 PASS$CLOSE: NOVALUE;                ! Called internally by RTL
67 0129 1
68 0130 1
69 0131 1 MACROS:
70 0132 1
71 0133 1 NONE
72 0134 1
73 0135 1 EQUATED SYMBOLS:
74 0136 1
75 0137 1 NONE
76 0138 1
77 0139 1 FIELDS:
78 0140 1
79 0141 1 NONE
80 0142 1
81 0143 1 OWN STORAGE:
82 0144 1
83 0145 1 NONE
```

```

85 0146 1 %SBTTL 'PASSCLOSE2 - Close a file'
86 0147 1 GLOBAL ROUTINE PASSCLOSE2 (
87 0148 1     PFV: REF $PASSPFV_FILE_VARIABLE,
88 0149 1     KEYWORDS: VECTOR [, LONG]
89 0150 1 ): NOVALUE =
90 0151 1
91 0152 1 ++
92 0153 1 FUNCTIONAL DESCRIPTION:
93 0154 1
94 0155 1     This routine closes a Pascal file. This entry is called from the
95 0156 1     compiled code for the CLOSE procedure.
96 0157 1
97 0158 1 CALLING SEQUENCE:
98 0159 1
99 0160 1     CALL PASSCLOSE2 (PFV.mr.r [, keyword.rz.v [, keyword.rz.v ...]])
100 0161 1
101 0162 1 FORMAL PARAMETERS:
102 0163 1
103 0164 1     PFV
104 0165 1         - The Pascal File Variable (PFV) passed by reference.
105 0166 1         The structure of the PFV is defined in PASPFV.REQ.
106 0167 1
107 0168 1     keyword
108 0169 1         - Zero or more values representing keywords in the
109 0170 1         CLOSE procedure. Some keywords are followed by
110 0171 1         associated arguments and values. See PASOPEKEY.REQ
111 0172 1         for the keyword value definitions.
112 0173 1
113 0174 1         If the keyword ERROR is specified, it must be first
114 0175 1         in the list, otherwise it will be ignored.
115 0176 1
116 0177 1 IMPLICIT INPUTS:
117 0178 1
118 0179 1     NONE
119 0180 1
120 0181 1 IMPLICIT OUTPUTS:
121 0182 1
122 0183 1     NONE
123 0184 1
124 0185 1 COMPLETION STATUS: (or ROUTINE VALUE:)
125 0186 1
126 0187 1     NONE
127 0188 1
128 0189 1 SIDE EFFECTS:
129 0190 1
130 0191 1     See PASS$CLOSE
131 0192 1
132 0193 1 SIGNALLED ERRORS:
133 0194 1
134 0195 1     FILALRCLO - file already closed
135 0196 1
136 0197 2 --
137 0198 2 BEGIN
138 0199 2
139 0200 2     LOCAL
140 0201 2         FCB: REF $PASSFCB_CONTROL_BLOCK,
141 0202 2         PFV_ADDR: VOLATILE,
142 0203 2         UNWIND_ACT: VOLATILE,
```

```

! Close a file
! File variable
! Keywords and specifiers
```

```

! File Control Block
! Enable argument
! Enable argument
```

PASSCLOSE2  
1-006

close a file  
PASSCLOSE2 - Close a file

K 14  
16-Sep-1984 01:29:07 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:51:21 [PASRTL.SRC]PASCLOSE2.B32;1

Page 4  
(3)

```

142 0203 2      ERROR_ADDR: VOLATILE;                ! Enable argument
143 0204 2
144 0205 2      BUILTIN
145 0206 2      ACTUALCOUNT,
146 0207 2      AP,
147 0208 2      CALLG;
148 0209 2
149 0210 2      ENABLE
150 0211 2      PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);
151 0212 2
152 0213 2      !+
153 0214 2      ! Set PFV_ADDR enable argument.
154 0215 2      !-
155 0216 2
156 0217 2      PFV_ADDR = PFV [PFV$R_PFV];
157 0218 2
158 0219 2      !+
159 0220 2      ! Validate and lock PFV.
160 0221 2      !-
161 0222 2
162 0223 2      PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
163 0224 2
164 0225 2      !+
165 0226 2      ! Set unwind action to unlock file.
166 0227 2      !-
167 0228 2
168 0229 2      UNWIND_ACT = PASS$UNWIND_UNLOCK;
169 0230 2
170 0231 2      !+
171 0232 2      ! If ERROR is present, it is the first keyword in the list. See
172 0233 2      ! if that is true. If so, set enable argument to the return address.
173 0234 2      !-
174 0235 2
175 0236 2      IF ACTUALCOUNT () GEQU 3      ! At least 3 needed for PFV, keyword, value
176 0237 2      THEN
177 0238 2          IF .KEYWORDS [0] EQL PASS$ERROR      ! Is it the ERROR keyword?
178 0239 2          THEN
179 0240 2              ERROR_ADDR = .KEYWORDS [1];
180 0241 2
181 0242 2      !+
182 0243 2      ! If file variable is not valid, call PASS$OPEN_IMPLICIT to see if the
183 0244 2      ! file should be implicitly opened. Even though we will just close it
184 0245 2      ! again, we must do this to satisfy the language semantics for INPUT
185 0246 2      ! and OUTPUT.
186 0247 2      !-
187 0248 2
188 0249 2      IF NOT .PFV [PFV$V_VALID]
189 0250 2      THEN
190 0251 2          PASS$OPEN_IMPLICIT (PFV [PFV$R_PFV], FCB [FCB$R_FCB]; FCB);
191 0252 2
192 0253 2      !+
193 0254 2      ! If FCB pointer is not valid, give an error.
194 0255 2      !-
195 0256 2
196 0257 2      IF NOT .PFV [PFV$V_FCB_VALID]
197 0258 2      THEN
198 0259 2          $PASS$IO_ERROR (PASS$_FILALRCLO,0);
```



PASSCLOSE2  
1-006

Close a file  
PASSCLOSE2 - Close a file

L 14  
16-Sep-1984 01:29:07  
14-Sep-1984 12:51:21

VAX-11 Bliss-32 V4.0-742  
[PASRTL.SRC]PASCLOSE2.B32;1

Page 5  
(3)

```
.. 199 0260 2
   200 0261 2
   201 0262 2
   202 0263 2
   203 0264 2
   204 0265 2
   205 0266 2
   206 0267 2
   207 0268 2
   208 0269 2
   209 0270 2
   210 0271 2
   211 0272 2
   212 0273 2
   213 0274 2
   214 0275 2
   215 0276 2
   216 0277 2
   217 0278 2
   218 0279 2
   219 0280 2
   220 0281 2
   221 0282 2
   222 0283 2
   223 0284 2
   224 0285 2
   225 0286 2
   226 0287 2
   227 0288 2
   228 0289 1

      + Zero the STATUS value in the FCB. This will get stored in the
      - PFV by PASS$REMOVE_FILE.

      FCB = .PFV [PFV$A_FCB];
      FCB [FCB$L_STATUS] = 0;

      + Call PASS$CLOSE to do the work.
      -

      CALLG (.AP, PASS$CLOSE);

      + Remove it from the list of open files.
      -

      PASS$REMOVE_FILE (FCB [FCB$R_FCB]);

      + Unlock the file variable.
      -

      PFV [PFV$V_LOCK] = 0;

      RETURN;

      END;
```

! End of routine PASS\$CLOSE2

```
.TITLE PASSCLOSE2 Close a file
.IDENT \1-006\

.EXTRN PASSCLOSE2, PASS$CLOSE
.EXTRN PASS$IO_HANDLER
.EXTRN PASS$VACIDATE PFV
.EXTRN PASS$OPEN_IMPLICIT
.EXTRN PASS$SIGNAL, PASS$FILALRCLO
.EXTRN PASS$REMOVE_FILE
```

```
.PSECT _PASSCODE, NOWRT, SHR, PIC, 2
```

```
SE 00C4 00000
08 08 C2 00002
7E D4 00005
AE 7C 00007
04 04 0000A
6D 005E CF DE 0000A
56 04 AC D0 0000F
08 AE 56 D0 00013
04 AE 0000G0000G 00 16 00017
03 AE 01 D0 0001D
03 6C 91 00021
19 08 0A 1F 00024
AC D1 00026
04 12 0002A
```

```
.ENTRY PASSCLOSE2, Save R2,R6,R7
SUBL2 #8, SP
CLRL ERROR_ADDR
CLRQ UNWIND_ACT
MOVAL 4$, (FP)
MOVL PFV, R6
MOVL R6, PFV_ADDR
JSB PASS$VACIDATE PFV
MOVL #1, UNWIND_ACT
CMPB (AP), #3
BLSSU 1$
CMPL KEYWORDS, #25
BNEQ 1$
```

```
.. 0147
   0197
   0217
   0223
   0229
   0236
   0239
   ..
```

PASSCLOSE2  
1-006

Close a file  
PASSCLOSE2 - Close a file

M 14  
16-Sep-1984 01:29:07 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:51:21 [PASRTL.SRC]PASCLOSE2.B32;1

Page 6  
(3)

	6E	0C	AC	D0	0002C		MOVL	KEYWORDS+4, ERROR_ADDR	:	0240
	06	06	A6	E8	00030	1\$:	BLBS	6(R6), 2\$	:	0249
		00000000G	00	16	00034		JSB	PASS\$OPEN_IMPLICIT	:	0251
OE	07	A6	06	E0	0003A	2\$:	BBS	#6, 7(R6), 3\$	:	0257
			7E	D4	0003F		CLRL	-(SP)	:	0259
	7E	00G	8F	9A	00041		MOVZBL	#PASSK.FILALRCLO, -(SP)	:	
00000000G	00		02	FB	00045		CALLS	#2, PASS\$\$SIGNAL	:	
				04	0004C		RET		:	
	57	0C	A6	D0	0004D	3\$:	MOVL	12(R6), FCB	:	0266
		D4	A7	D4	00051		CLRL	-44(FCB)	:	0267
0000V	CF		6C	FA	00054		CALLG	(AP), PASS\$CLOSE	:	0273
			57	DD	00059		PUSHL	FCB	:	0279
00000000G	00		01	FB	0005B		CALLS	#1, PASS\$REMOVE_FILE	:	
	50	04	AC	D0	00062		MOVL	PFV, R0	:	0285
07	A0	80	8F	8A	00066		BICB2	#128, 7(R0)	:	
				04	0006B		RET		:	0289
				0000	0006C	4\$:	.WORD	Save nothing	:	0197
	50	08	AC	D0	0006E		MOVL	8(AP), R0	:	
	50	04	A0	D0	00072		MOVL	4(R0), R0	:	
		F4	A0	9F	00076		PUSHAB	ERROR_ADDR	:	
		F8	A0	9F	00079		PUSHAB	UNWIND_ACT	:	
		FC	A0	9F	0007C		PUSHAB	PFV_ADDR	:	
			03	DD	0007F		PUSHL	#3	:	
			5E	DD	00081		PUSHL	SP	:	
	7E	04	AC	7D	00083		MOVQ	4(AP), -(SP)	:	
00000000G	00		03	FB	00087		CALLS	#3, PASS\$IO_HANDLER	:	
				04	0008E		RET		:	

; Routine Size: 143 bytes, Routine Base: \_PASS\$CODE + 0000

: 229 0290 1  
: 230 0291 1 !<BLF/PAGE>



```
232 0292 1 $SBTTL 'PASS$CLOSE - Close a file'
233 0293 1 GLOBAL ROUTINE PASS$CLOSE (
234 0294 1     IN PFV: REF $PASS$PFV_FILE_VARIABLE,
235 0295 1     KEYWORDS: VECTOR [, [LONG]]
236 0296 1 ): NOVALUE =
237 0297 1
238 0298 1 !++
239 0299 1 ! FUNCTIONAL DESCRIPTION:
240 0300 1
241 0301 1     This procedure closes the Pascal file specified by 'PFV'
242 0302 1     using the attributes possibly supplied as 'keyword'. PASS$CLOSE
243 0303 1     is to be called from other RTL routines only after they have tested
244 0304 1     and set PFV$V_LOCK in 'PFV'.
245 0305 1
246 0306 1 ! CALLING SEQUENCE:
247 0307 1
248 0308 1     PASS$CLOSE (PFV.mr.r [, keyword.rz.v [, keyword.rz.v ...]])
249 0309 1
250 0310 1 ! FORMAL PARAMETERS:
251 0311 1
252 0312 1     PFV           - The Pascal File Variable (PFV) passed by reference.
253 0313 1                   The structure of the PFV is defined in PASPFV.MDL.
254 0314 1
255 0315 1     keyword       - Zero or more values representing keywords in the
256 0316 1                   CLOSE procedure. Some keywords are followed by
257 0317 1                   associated arguments and values. See PASOPNKEY.MDL
258 0318 1                   for the keyword value definitions.
259 0319 1
260 0320 1 ! IMPLICIT INPUTS:
261 0321 1
262 0322 1     The PFV$V_LOCK bit is assumed to have been tested-and-set by the caller.
263 0323 1     The PFV$V_FCB_VALID bit is assumed set.
264 0324 1
265 0325 1 ! IMPLICIT OUTPUTS:
266 0326 1
267 0327 1     NONE
268 0328 1
269 0329 1 ! COMPLETION STATUS:
270 0330 1
271 0331 1     NONE
272 0332 1
273 0333 1 ! SIDE EFFECTS:
274 0334 1
275 0335 1     The file is closed. The record buffer and resultant name string
276 0336 1     are deallocated.
277 0337 1
278 0338 1 ! SIGNALLED ERRORS:
279 0339 1
280 0340 1     ERRDURCLO - error during CLOSE
281 0341 1     INVARGPAS - invalid argument to Pascal Run-Time Library
282 0342 1     FILNAMREQ - file name required to save, submit or print file
283 0343 1
284 0344 1 !--
285 0345 1
286 0346 2 BEGIN
287 0347 2
288 0348 2 LOCAL
```

```
289 0349 2 PFV: REF $PASS$PFV_FILE VARIABLE, Pascal File Variable
290 0350 2 FCB: REF $PASS$FCB_CONTROL_BLOCK, File control block
291 0351 2 BLOCK_ADDR: VOLATILE, Address of dynamically allocated block
292 0352 2 STATUS, Status from $CLOSE
293 0353 2 KEYWORDS_SEEN: BITVECTOR [PASS$K_OPEKEYHI+1], Keep track of keywords seen
294 0354 2 USER_ACTION_BPV: REF VECTOR [, LONG]; USER_ACTION routine BPV
295 0355 2
296 0356 2
297 0357 2 BUILTIN
298 0358 2 TESTBITCS,
299 0359 2 TESTBITSC,
300 0360 2 ACTUALCOUNT;
301 0361 2
302 0362 2 BIND
303 0363 2 RAB = FCB: REF BLOCK [, BYTE],
304 0364 2 FAB = FCB: REF $PASS$FAB_FCB_STRUCT;
305 0365 2
306 0366 2 !+
307 0367 2 !- Get PFV address.
308 0368 2 !-
309 0369 2
310 0370 2 PFV = .IN_PFV;
311 0371 2
312 0372 2 !+
313 0373 2 !- Get addresses of PFV.
314 0374 2 !-
315 0375 2
316 0376 2 FCB = .PFV [PFV$A_FCB];
317 0377 2
318 0378 2 !+
319 0379 2 !- Do the keyword processing.
320 0380 2 !-
321 0381 2
322 0382 2 KEYWORDS_SEEN <0,32> = 0; ! Indicate no keywords seen
323 0383 2 KEYWORDS_SEEN <32,PASS$K_OPEKEYHI+-31> = 0;
324 0384 2
325 0385 2
326 0386 2 !+
327 0387 2 !- Look through keyword list and set all specified attributes.
328 0388 2 !-
329 0389 2
330 0390 2 INCR I FROM 0 TO (ACTUALCOUNT () - 2) DO
331 0391 2 BEGIN
332 0392 2
333 0393 2 LOCAL
334 0394 2 KEYWD_VALUE;
335 0395 2
336 0396 2 !+
337 0397 2 !- Check for valid keyword code. If not already seen, use the
338 0398 2 keyword.
339 0399 2 !-
340 0400 2
341 0401 2 KEYWD_VALUE = .KEYWORDS [I];
342 0402 2 IF .KEYWD_VALUE LSS PASS$K_OPEKEYLO OR .KEYWD_VALUE GTR PASS$K_OPEKEYHI
343 0403 2 THEN
344 0404 2 $PASS$IO_ERROR (PASS$_INVARGPAS,0);
345 0405 2
```

```
346 0406 3 IF TESTBITS (KEYWORDS_SEEN [.PAS$AB_KEYWD_NAME_TABLE[.KEYWD_VALUE]])
347 0407 3 THEN
348 0408 3 CASE .KEYWD_VALUE FROM PAS$K_OPEKEYLO TO PAS$K_OPEKEYHI OF
349 0409 3 SET
350 0410 3
351 0411 3 [PAS$K_DISPOSITION_SAVE]:
352 0412 3 BEGIN
353 0413 3 FCB [FCB$V_DELETE] = 0;
354 0414 3 FCB [FCB$V_PRINT] = 0;
355 0415 3 FCB [FCB$V_SUBMIT] = 0;
356 0416 3 FCB [FCB$V_SAVE] = 1;
357 0417 3 END;
358 0418 3
359 0419 3 [PAS$K_DISPOSITION_DELETE]:
360 0420 3 BEGIN
361 0421 3 FCB [FCB$V_PRINT] = 0;
362 0422 3 FCB [FCB$V_SUBMIT] = 0;
363 0423 3 FCB [FCB$V_SAVE] = 0;
364 0424 3 FCB [FCB$V_DELETE] = 1;
365 0425 3 END;
366 0426 3
367 0427 3 [PAS$K_DISPOSITION_PRINT]:
368 0428 3 BEGIN
369 0429 3 FCB [FCB$V_DELETE] = 0;
370 0430 3 FCB [FCB$V_SUBMIT] = 0;
371 0431 3 FCB [FCB$V_SAVE] = 1;
372 0432 3 FCB [FCB$V_PRINT] = 1;
373 0433 3 END;
374 0434 3
375 0435 3 [PAS$K_DISPOSITION_PRINT_DELETE]:
376 0436 3 BEGIN
377 0437 3 FCB [FCB$V_SUBMIT] = 0;
378 0438 3 FCB [FCB$V_SAVE] = 0;
379 0439 3 FCB [FCB$V_PRINT] = 1;
380 0440 3 FCB [FCB$V_DELETE] = 1;
381 0441 3 END;
382 0442 3
383 0443 3 [PAS$K_DISPOSITION_SUBMIT]:
384 0444 3 BEGIN
385 0445 3 FCB [FCB$V_PRINT] = 0;
386 0446 3 FCB [FCB$V_DELETE] = 0;
387 0447 3 FCB [FCB$V_SUBMIT] = 1;
388 0448 3 FCB [FCB$V_SAVE] = 1;
389 0449 3 END;
390 0450 3
391 0451 3 [PAS$K_DISPOSITION_SUBMIT_DELETE]:
392 0452 3 BEGIN
393 0453 3 FCB [FCB$V_PRINT] = 0;
394 0454 3 FCB [FCB$V_SAVE] = 0;
395 0455 3 FCB [FCB$V_SUBMIT] = 1;
396 0456 3 FCB [FCB$V_DELETE] = 1;
397 0457 3 END;
398 0458 3
399 0459 3 [PAS$K_USER_ACTION]:
400 0460 3 BEGIN
401 0461 3 USER_ACTION_BPV = .KEYWORDS [(I=.I+1)];
402 0462 3 END;
```



```
403 0463 3
404 0464 3
405 0465 4
406 0466 4
407 0467 4
408 0468 4
409 0469 4
410 0470 4
411 0471 3
412 0472 3
413 0473 3
414 0474 3
415 0475 3
416 0476 3
417 0477 3
418 0478 3
419 0479 3
420 0480 3
421 0481 3
422 0482 3
423 0483 3
424 0484 3
425 0485 3
426 0486 3
427 0487 3
428 0488 3
429 0489 3
430 0490 3
431 0491 3
432 0492 3
433 0493 3
434 0494 3
435 0495 3
436 0496 3
437 0497 3
438 0498 3
439 0499 3
440 0500 3
441 0501 3
442 0502 3
443 0503 3
444 0504 3
445 0505 3
446 0506 3
447 0507 3
448 0508 3
449 0509 3
450 0510 3
451 0511 3
452 0512 3
453 0513 3
454 0514 3
455 0515 3
456 0516 3
457 0517 3
458 0518 3
459 0519 2

[PASS$K_ERROR]:
BEGIN
+ The ERROR parameter, if specified, was already processed
- in PASS$CLOSE, so just ignore it here.
I = .I + 1; ! Ignore next parameter
END;

[INRANGE,OUTRANGE]:
$PASS$IO_ERROR (PASS$_INVARGPAS,0);

TES;

END;

+ Check for incompatible DISPOSITION.
-

IF .FAB [FAB$V_TMD] AND
(.FCB [FCB$V_SAVE] OR .FCB [FCB$V_SUBMIT] OR .FCB [FCB$V_PRINT])
THEN
$PASS$IO_ERROR (PASS$_FILNAMREQ,0); ! File name required

+ Fill in fields in RMS control blocks
-

IF .FCB [FCB$V_DELETE] AND NOT .FAB [FAB$V_TMD]
THEN
FAB [FAB$V_DLT] = 1;

IF .FCB [FCB$V_SUBMIT]
THEN
FAB [FAB$V_SCF] = 1;

IF .FCB [FCB$V_PRINT]
THEN
FAB [FAB$V_SPL] = 1;

+ If a textfile, and if in Generation mode, and if the buffer
+ is not empty, do a WRITELN.
-

IF .FCB [FCB$V_TEXT] AND .FCB [FCB$V_GENERATION] AND
(.FCB [FCB$A_RECORD_CUR] GT .FCB [FCB$A_RECORD_BEG])
THEN
PASS$WRITELN (PFV [PFV$R_PFV], FCB [FCB$R_FCB]);

+ Make sure the XAB chain doesn't point to garbage.
-

FAB [FAB$L_XAB] = 0;
```

```
460 0520 2
461 0521 2
462 0522 2
463 0523 2
464 0524 2
465 0525 2
466 0526 2
467 0527 2
468 0528 2
469 0529 2
470 0530 2
471 0531 2
472 0532 2
473 0533 2
474 0534 2
475 0535 2
476 0536 2
477 0537 2
478 0538 2
479 0539 2
480 0540 2
481 0541 2
482 0542 2
483 0543 2
484 0544 2
485 0545 2
486 0546 2
487 0547 2
488 0548 2
489 0549 2
490 0550 2
491 0551 2
492 0552 2
493 0553 2
494 0554 2
495 0555 2
496 0556 2
497 0557 2
498 0558 2
499 0559 2
500 0560 2
501 0561 2
502 0562 2
503 0563 2
504 0564 2
505 0565 2
506 0566 2
507 0567 2
508 0568 2
509 0569 2
510 0570 2
511 0571 2
512 0572 2
513 0573 2
514 0574 2
515 0575 2
516 0576 2

!+
!- Close the file.
!-
IF .FAB [FAB$W_IFI] NEQ 0
THEN
  BEGIN
    IF .KEYWORDS_SEEN [PAS$K_USER_ACTION]
    THEN
      BEGIN
        LINKAGE
          USER_ACTION_LNK = CALL (REGISTER=1, STANDARD, STANDARD);

        STATUS = USER_ACTION_LNK (.USER_ACTION_BPV [0],      ! Address
                                   .USER_ACTION_BPV [1],      ! Environment
                                   FAB [0,0,0,0],              ! FAB address
                                   RAB [0,0,0,0],              ! RAB address
                                   PFV [PFV$R_PFV]);            ! PFV address
      END
    ELSE
      STATUS = $PAS$RMS_OP ($CLOSE (FAB=FAB [0,0,0,0]));

    !+
    !- Set the IFI to zero so we don't try another $CLOSE if the last one
    !- failed.
    FAB [FAB$W_IFI] = 0;

    !+
    !- If the close failed, signal an error.
    !-
    IF NOT .STATUS
    THEN
      $PAS$IO_ERROR (PAS$_ERRDURCLO);

    END;

    !+
    !- Deallocate the record buffer, if necessary.
    !-
    IF TESTBITSC (FCB [FCB$V_DYNAMIC_UBF])
    THEN
      PAS$$FREE_VM (.RAB [RAB$W_USZ], RAB [RAB$L_UBF]);

    !+
    !- Deallocate key types vector, if necessary.
    !-
    IF .FCB [FCB$A_KEY_TYPES] NEQ 0
    THEN
      PAS$$FREE_VM (.FCB [FCB$L_NKEYS]*4, FCB [FCB$A_KEY_TYPES]);

    !+
    !- Deallocate resultant name string, if dynamic. This should be the
```

PAS\$CLOSE2  
1-006

close a file  
PAS\$\$CLOSE - Close a file

F 15  
16-Sep-1984 01:29:07  
14-Sep-1984 12:51:21

VAX-11 Bliss-32 V4.0-742  
[PASRTL.SRC]PAS\$CLOSE2.B32;1

Page 12  
(4)

```

517      0577      2      ! last thing to do.
518      0578      2      !-
519      0579      2
520      0580      2      IF TESTBITSC (FCB [FCB$V_DYNAMIC_RSN])
521      0581      2      THEN
522      0582      2          BEGIN
523      0583      2              LOCAL
524      0584      2                  NAM: REF BLOCK [, BYTE];
525      0585      2                  NAM = .FAB [FAB$L_NAM];
526      0586      2                  PASS$FREE_VM (.NAM [NAM$B_RSL], NAM [NAM$L_RSA]);
527      0587      2                  END;
528      0588      2
529      0589      2      !+
530      0590      2      ! Mark PFV to indicate file closed.
531      0591      2      !-
532      0592      2
533      0593      2      PFV [PFV$V_OPEN] = 0;
534      0594      2      PFV [PFV$V_VALID] = 0;      ! Invalidate flags
535      0595      2      PFV [PFV$V_DFB] = 0;      ! Invalidate file buffer
536      0596      2      PFV [PFV$V_EOF_DEFINED] = 0;
537      0597      2
538      0598      2      RETURN;
539      0599      2
540      0600      1      END;

```

! End of routine PAS\$\$CLOSE

```

.EXTRN  PASSK_INVARGPAS
.EXTRN  PASS$AB KEYWD_NAME_TABLE
.EXTRN  PASSK_FILNAMREQ
.EXTRN  PASS$WRITELN, SY$CLOSE
.EXTRN  PASSK_ERRDURCLO
.EXTRN  PASS$FREE_VM

```

			OFFC	00000				
		5B	00000000G	00	9E	00002		
		5A	00000000G	00	9E	00009		
		5E		08	C2	00010		
		56	04	AC	D0	00013		
		57	0C	A6	D0	00017		
				7E	D4	0001B		
	04	AE	01FF	8F	AA	0001D		
		54		6C	9A	00023		
		54		02	C2	00026		
		52		01	CE	00029		
				00C8	31	0002C	1\$:	
		53	08	AC42	D0	0002F	2\$:	
				65	15	00034		
		28		53	D1	00036		
				60	14	00039		
		50	00000000G	0043	9A	0003B		
		6E		50	E2	00043		
		01		53	CF	00047		
0050	0050	0050		0050		0004B	3\$:	
0050	0050	0050		0050		00053		
0050	0050	0050		0050		0005B		

  

		.ENTRY	PAS\$\$CLOSE, Save R2,R3,R4,R5,R6,R7,R8,R9,-				0293	
			R10,R11					
		MOVAB	PAS\$\$SIGNAL, R11					
		MOVAB	PAS\$FREE_VM, R10					
		SUBL2	#8, SP					
		MOVL	IN PFV, PFV				0370	
		MOVL	12(PFV), FCB				0376	
		CLRL	KEYWORDS_SEEN				0382	
		BICW2	#511, KEYWORDS_SEEN+4				0383	
		MOVZBL	(AP), R4				0390	
		SUBL2	#2, R4					
		MNEGL	#1, I					
		BRW	15\$					
		MOVL	KEYWORDS[I], KEYWD_VALUE				0401	
		BLEQ	4\$				0402	
		CMPL	KEYWD_VALUE, #40					
		BGTR	4\$					
		MOVZBL	PAS\$AB KEYWD_NAME_TABLE[KEYWD_VALUE], R0				0406	
		BBSS	R0, KEYWORDS_SEEN, -1\$					
		CASEL	KEYWD_VALUE, -#1, #39				0408	
		.WORD	4\$-3\$,-					
			4\$-3\$,-					
			4\$-3\$,-					



0474  
0415  
0416  
0423  
0424  
0430  
0432  
0408  
0437  
0439  
0440  
0446  
0447  
0448  
0408  
0454  
0455  
0456



PASSCLOSE2  
1-006

Close a file  
PASS\$CLOSE - Close a file

I 15  
16-Sep-1984 01:29:07  
14-Sep-1984 12:51:21

VAX-11 Bliss-32 V4.0-742  
[PASRTL.SRC]PASCLOSE2.B32;1

Page 15  
(4)

7E	D0	A7	CC	A7	9F	001B0	PUSHAB	-52(FAB)	:	0573
		6A		02	78	001B3	ASHL	#2, -48(FAB), -(SP)	:	
0E		68		02	FB	001B8	CALLS	#2, PASS\$FREE_VM	:	
		50		12	E5	001BB	BBCC	#18, (R8), 28\$	:	0580
			5C	A7	D0	001BF	MOVL	108(FAB), NAM	:	0585
		7E	04	A0	9F	001C3	PUSHAB	4(NAM)	:	0586
		6A	03	A0	9A	001C6	MOVZBL	3(NAM), -(SP)	:	
06	A6			02	FB	001CA	CALLS	#2, PASS\$FREE_VM	:	
			2007	8F	AA	001CD	BICW2	#8199, 6(PFV)-VM	:	0596
				04	001D3		RET		:	0600

; Routine Size: 468 bytes, Routine Base: \_PASS\$CODE + 008F

: 541 0601 1  
: 542 0602 1 !<BLF/PAGE>



PASSCLOSE2  
1-006

Close a file  
PASS\$CLOSE - Close a file

J 15  
16-Sep-1984 01:29:07  
14-Sep-1984 12:51:21

VAX-11 Bliss-32 V4.0-742  
[PASRTL.SRC]PASCLOSE2.B32;1

Page 16  
(5)

: 544 0603 1 END  
: 545 0604 1  
: 546 0605 0 ELUDOM

! End of module PASS\$CLOSE2

# PSECT SUMMARY

Name	Bytes	Attributes
_PASS\$CODE	611	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

# Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	16	0	581	00:01.0
-\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	111	25	33	00:00.4

# COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:PASCLOSE2/OBJ=OBJ\$:PASCLOSE2 MSRC\$:PASCLOSE2/UPDATE=(ENH\$:PASCLOSE2)

: Size: 611 code + 0 data bytes  
: Run Time: 00:14.1  
: Elapsed Time: 00:51.1  
: Lines/CPU Min: 2578  
: Lexemes/CPU-Min: 19367  
: Memory Used: 193 pages  
: Compilation Complete



0293 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

